

#5 M

WEST

☐ Generate Collection

L10: Entry 2 of 11

File: JPAB

Jun 23, 2000

PUB-NO: JP02000174794A

DOCUMENT-IDENTIFIER: JP 2000174794 A

TITLE: DEVICE AND METHOD FOR CONTROLLING COMMUNICATION

PUBN-DATE: June 23, 2000

INVENTOR-INFORMATION:

NAME

COUNTRY

SENDA, MAKOTO

N/A

ASSIGNEE-INFORMATION:

NAME

COUNTRY

CANON INC

N/A

APPL-NO: JP10350573

APPL-DATE: December 9, 1998

INT-CL (IPC): H04L 12/44; G06F 13/14; H04L 29/00

ABSTRACT:

PROBLEM TO BE SOLVED: To reconstruct a network by automatically canceling a loop configuration by disabling one of ports related to a loop when the duration of a Parent Notify signal is measured and that loop is detected.

SOLUTION: At the point of time the loop is detected at respective nodes (nods 1 to 3 and 5) composing of the loop, the timer of certain arbitrary time set for each node is activated at the respective nodes. Since the node 1 is first made into time-out, the port connected with the node 2 is disabled, a route is cut and bus reset is generated. After the bus is completely initialized, the Parent Notify signal from the node 1, which changes from 'Branch' node to 'Leaf' node, and nodes 4 and 6 to 8 of 'Leaf' nodes to the node of a parent candidate is transmitted to the port. The nodes 2, 3 and 5 of 'Branch' nodes transmit Child Notify signals to the node, which receives the Parent Notify signal, for confirming a child node.

COPYRIGHT: (C)2000,JPO

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-174794

(P2000-174794A)

(13) 公開日 平成12年6月23日 (2000.6.23)

(51) Int.Cl. ⁷	識別記号	F I	マーク	マーク (参考)
H 0 4 L 12/44		H 0 4 L 11/00	3 4 0	5 B 0 1 4
G 0 6 F 13/14	3 3 0	G 0 6 F 13/14	3 3 0 Z	5 K 0 3 3
H 0 4 L 29/00		H 0 4 L 13/00	Z	5 K 0 3 4

審査請求 未請求 請求項の数 9 O L (全 16 頁)

(21) 出願番号 特願平10-350573

(22) 出願日 平成10年12月9日 (1998.12.9)

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 千田 誠

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

(74) 代理人 100076428

弁理士 大塚 康德 (外2名)

Fターム (参考) 5B014 HC13

5K033 AA05 AA06 DA11 DA15 DB17

DB21 EA02 EA05 EA06 EC01

5K034 AA05 AA16 CC01 DD06 HH04

HH07 HH13 HH63 LL03 LL08

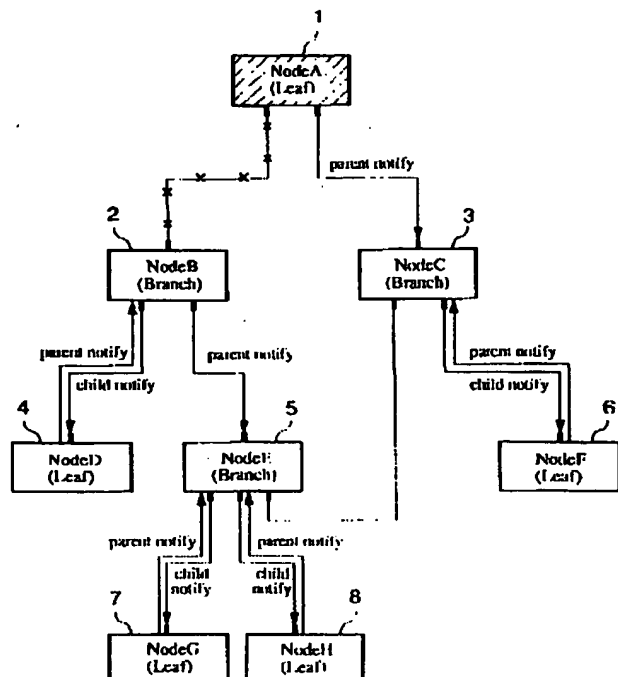
TT01

(54) 【発明の名称】 通信制御装置及び方法

(57) 【要約】

【課題】 IEEE1394ネットワーク中のループを検出し、それを自動的に解除する。

【解決手段】 バスリセットが発生すると、複数のポートで他のノードと接続されているノードは、最後のひとつのポートを残して他のポートからParent Notifyの受信を待ち、最後のポートからParent Notifyを送信する。この際に、所定時間以上複数のポートからParent Notifyを受けられないと、ループがあると判定されて、ひとつのリンクが切断され、改めてバスリセットされる。



【特許請求の範囲】

【請求項1】 互いに従属関係を通知し合ってデバイス同士の接続関係を構成する通信制御装置であって、少なくとも1つのポートを介して他のデバイスと接続する接続手段と、

他のデバイスとの接続関係を構成中に前記接続手段に、接続されたデバイスがループを構成していることを検出するループ検出手段と、

前記ループ検出手段によりループが検出された場合、所定時間後にループが検出されたポートの少なくとも1つをディセーブルにするとともに、デバイス同士の接続関係を構成し直すためのリセット通知を他のデバイスに通知するディセーブル手段とを備えることを特徴とする通信制御装置。

【請求項2】 前記所定時間は、ループを形成しているポート数がより多いデバイスにはより長く設定されることを特徴とする請求項1に記載の通信制御装置。

【請求項3】 前記所定時間は、ループを形成しているポート数に比例した第一の値と、デバイス固有でかつループを形成するポート一つ分の時間よりも短い第二の値とを加算した時間として設定されることを特徴とする請求項1または2に記載の通信制御装置。

【請求項4】 前記ループ検出手段は、前記リセット通知を受けた場合、あるいはリセット通知を出力した場合に、ポートの数が複数であれば、そのポートのうちのいずれか1つを残して、他のポートを介して接続されたデバイスから該デバイスが子ノードである旨の通知の受信を所定時間待ち、その所定時間を経過した場合にループの存在を検出することを特徴とする請求項1乃至3のいずれかに記載の通信制御装置。

【請求項5】 少なくとも1つのポートを介して他のデバイスと接続され、互いに従属関係を通知し合って互いの接続関係を構成するデバイスのための通信制御方法であって、

他のデバイスとの接続関係を構成中に前記接続手段に、接続されたデバイスがループを構成していることを検出するループ検出工程と、

前記ループ検出工程によりループが検出された場合、所定時間後にループが検出されたポートの少なくとも1つをディセーブルにするとともに、デバイス同士の接続関係を構成し直すためのリセット通知を他のデバイスに通知するディセーブル工程とを備えることを特徴とする通信制御方法。

【請求項6】 前記所定時間は、ループを形成しているポート数がより多いデバイスにはより長く設定されることを特徴とする請求項5に記載の通信制御方法。

【請求項7】 前記所定時間は、ループを形成しているポート数に比例した第一の値と、デバイス固有でかつループを形成するポート一つ分の時間よりも短い第二の値とを加算した時間として設定されることを特徴とする請

求項5または6に記載の通信制御方法。

【請求項8】 前記ループ検出工程は、前記リセット通知を受けた場合、あるいはリセット通知を出力した場合に、ポートの数が複数であれば、そのポートのうちのいずれか1つを残して、他のポートを介して接続されたデバイスから該デバイスが子ノードである旨の通知の受信を所定時間待ち、その所定時間を経過した場合にループの存在を検出することを特徴とする請求項5乃至7のいずれかに記載の通信制御方法。

10 【請求項9】 少なくとも1つのポートを介して接続され、互いに従属関係を通知し合って互いの接続関係を構成するコンピュータにより実行可能なプログラムを記憶する記憶媒体であって、前記プログラムは、他のデバイスとの接続関係を構成中に前記接続手段に、接続されたデバイスがループを構成していることを検出するループ検出工程と、

前記ループ検出工程によりループが検出された場合、所定時間後にループが検出されたポートの少なくとも1つをディセーブルにするとともに、デバイス同士の接続関係を構成し直すためのリセット通知を他のデバイスに通知するディセーブル工程とを含むことを特徴とする記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、例えばシリアルバス等を用いた通信制御装置及び通信制御方法に関する。

【0002】

30 【従来の技術】近年、シリアルバスが、信号線が少ないこと、ケーブルが細いこと、コネクタが小さいこと、IDやターミネータ等の設定が不要なこと、活線挿抜が可能なこと、等時性のあるデータ転送が可能なこと等の特徴をもつことから脚光を浴びている。

【0003】特に、IEEE1394のシリアルバスは、動画像等の大容量データを高速伝送可能であること、バスアーキテクチャによりメモリアクセスが可能であること、またピア・ツー・ピア接続が可能であることなどの特徴があり、パーソナルコンピュータだけでなく、家庭内のAV機器やそれ以外の機器への適用が盛んに進められている。

40 【0004】また、更なる高速化や長距離化に対応した新たな規格も現在進められている。

【0005】IEEE1394のバス構成例を図6に示す。

50 【0006】1～8は、IEEE1394インターフェースでデジーチェーン接続あるいはブランチ接続で接続されている装置（ノード）である。IEEE1394ポートは、2組のツイストペアケーブル（一方をA、他方をBと称す）と1組の電源ペアケーブルの計6本のケーブルで構成され、2組のツイストペアケーブルはお互いケーブルでクロスして接続され、一方のAは他方のBに、一方のBは他方のAに接続される。

【0007】データ信号は、半二重通信で、ツイストペアAでデータ信号をツイストペアBでストロブ信号を差動信号として送信し、ツイストペアAでストロブ信号をツイストペアBでデータ信号を作動信号として受信する。また、調停信号は、前二重通信で、ドライバ側から調停信号を送信し、同時にレシーバ側ではある電圧レベルによって定義された3相(0, 1, Z)の論理を用いて受信した調停信号を識別する。

【0008】上記の3相の論理を判別するために、ツイストペアAには「Arb A Rx」として、ツイストペアBには「Arb B Rx」として非反転入力と反転入力を1組にしたコンパレータが用意されている。「0」は非反転入力での低電圧と反転入力での高電圧と定義され、「1」は非反転入力での高電圧と反転入力での低電圧と定義され、「Z」は非反転と反転の両入力での低電圧で定義されている。

【0009】バス構成を確立するには、バスの初期化、ツリー識別、自己識別の3つのフェーズが実施される。この3つのフェーズのプロセスの間に、図のようなツリーに似たトポロジが構築される。

【0010】ここで、バスの初期化は、電源不連続で発生する場合や、ソフトコマンドによって発生する場合や、または、バス構成でノードの新規接続や切り離し等によるバス構成の不調を検出した場合によりバスリセットが発生し、すべてのノードがアイドル状態になることにより完了する。

【0011】具体的には、まずツイストペアAとツイストペアBの両方に論理「1」を送信してバスリセットを発生し、この信号を受信したノードはバスリセットの発生を認識する。バスリセット発生後任意の時間が経過すると、ツイストペアAとツイストペアBの両方に論理「Z」を送信してアイドル信号を発生させ、接続されている相手ノードがアイドル信号を送信するのを待ち、相手ノードからアイドル信号を受信するとバスの初期化が完了したことを認識する。

【0012】バスの初期化が完了すると、次にツリー識別に移る。

【0013】ここで、ノードには、1つしか接続ノードを持たないノードを「Leaf」ノードと、2以上の接続ノードを持つノードを「branch」ノードと、接続ノードを持たない「非接続」ノードの3種類のノードがある。

【0014】ツリー識別では、まず最初に、ノードが親(Root)候補であるノードに対してParent notify信号を送信する。図6では、「Leaf」ノード4, 6, 7, 8からParent notify信号がポートに送信される。「Branch」ノードでは、複数ポートある場合に、どのポートがRoot候補のノードであるか認識することができないので最初にParent notify信号を送信することはなく、「Leaf」ノードからのParent notify信号を受信するのを待つ。

【0015】ただし、「Branch」ノードで複数ポートを有していてもノードが接続されているポートが1ポートしかなく他のポートはすべて非接続の場合には「Leaf」ノードになる。

【0016】次に、図7は、「Branch」ノード3, 5がParent notify信号を受信したノードに対して子ノードであることを認めるChild notify信号を送信し、「Leaf」ノードはChild notify信号を受信しそのポートの接続が完了したことを認識する。

【0017】「Branch」ノードは、上述したように、Parent notify信号に対してChild notify信号を送信し接続が完了した後にParent notify信号を受信していないポートが1ポートになると、そのポートに対してはParent notify信号を送信し、Child notify信号を受信するとこのポートのバス構成の接続が完了するが、図7は「Branch」ノード3と5が上記により接続を完了したことを示している。

【0018】図8は「Branch」ノード1からParent notify信号を送信し、「Branch」ノード2からChild notify信号を送信「Root」ノードがノード2に確定しツリー識別が完了したことを示している。

【0019】ツリー識別が完了すると、次に各ノードにノード番号を割り振りお互いがデータ通信を可能にするための自己識別に入る。

【0020】

【発明が解決しようとする課題】しかしながら、上述した従来の技術においては、コネクタがすべて同一であることから、図9に示すようにノード3とノード5が接続されると、デージェチェーン接続やブランチ接続以外の許容されていないループ接続がノード1とノード2のノード3とノード5との間で成立してしまう。

【0021】このループ接続では、「Branch」ノードの複数ポートが、他ノードからのparent notify信号を受信するのを待つしまうため、この複数ポートからParent notify信号を送信することはなくツリー識別を完了させることができないため、その後の自己識別やデータ通信ができない状態に陥ってしまうという問題がある。

【0022】本発明は上記従来例に鑑みてなされたもので、ループを検出してループ構成を自動的に解消してネットワークを再構築する通信制御装置及び方法を提供することを目的とする。

【0023】

【課題を解決するための手段】上記目的を達成するために、本発明は、Parent Notify信号の待ち時間を測定してループを検出し、ループが検出されたならそのループにかかるポートの少なくとも1つをディセーブルにしてネットワークを再構築することで、ループを解消する。

【0024】あるいは、次のような構成からなる。すなわち、互いに従属関係を通知し合ってデバイス同士の接続関係を構成する通信制御装置であって、少なくとも1

つのポートを介して他のデバイスと接続する接続手段と、他のデバイスとの接続関係を構成中に前記接続手段に、接続されたデバイスがループを構成していることを検出するループ検出手段と、前記ループ検出手段によりループが検出された場合、所定時間後にループが検出されたポートの少なくとも1つをディセーブルにするとともに、デバイス同士の接続関係を構成し直すためのリセット通知を他のデバイスに通知するディセーブル手段とを備える。

【0025】また望ましくは、前記所定時間は、ループを形成しているポート数がより多いデバイスにはより長く設定される。

【0026】また望ましくは、前記所定時間は、ループを形成しているポート数に比例した第一の値と、デバイス固有でかつループを形成するポート一つ分の時間よりも短い第二の値とを加算した時間として設定される。

【0027】また望ましくは、前記ループ検出手段は、前記リセット通知を受けた場合、あるいはリセット通知を出力した場合に、ポートの数が複数であれば、そのポートのうちのいずれか1つを残して、他のポートを介して接続されたデバイスから該デバイスが子ノードである旨の通知の受信を所定時間待ち、その所定時間を経過した場合にループの存在を検出する。

【0028】

【発明の実施の形態】図1において、ノード1～8は、IEEE1394インターフェースで接続されている装置（ノード）である。本発明では、各ノードはループ検出機能を持ち、各ノードが有するIEEE1394の各ポートを独立してディセーブルさせる機能を持ち、各ノードは任意の時間タイマ機能をもつ。

【0029】図5は、ノードの一例であるコンピュータのブロック図である。IEEE1394インターフェース501により、ケーブルを介して他のデバイスと接続される。図5のインターフェースは3つのポートを備えている。HDD502には、データやプログラムファイルが格納されており、RAM505にロードされてCPU504により処理あるいは実行される。CDROMやフロッピーディスク、光磁気ディスク等の取り外し可能な媒体を用いるファイル装置503には、データあるいはプログラムを記録した媒体がセットされ、それらはHDD502に複写されたりRAM505にロードされる。このプログラムやデータには、後述するIEEE1394によるデータの送受信やバスリセット時の処理、ループの検出といった処理を行うためのものが含まれる。ROM506にもプログラムやデータが記憶されている。タイマ1, 2, 3は、後述するバスリセット時のループ検出及び解消のために用いられる。なお、図5はコンピュータの構成であるが、IEEE1394には、プリンタなどの入出力デバイス等も接続できる。このようなデバイスでは、図5の構成に加えて入出力のための機構が備えられ

ている。

【0030】次に、ノードの動作説明をする。各ノードは上記機能を用いて、ノードがループを検出すると任意の時間タイマを起動し、タイマアウトが発生した場合には、自ノードのループを検出した複数ポートのうちのいずれか1ポートをディセーブルさせ、バスリセットを発生してバスの初期化から再スタートさせる。もしタイマアウトする前にバスの初期化が開始された場合には、タイマをストップしてバスの初期化から再スタートする。

【0031】図1では、上記動作の具体例を示している。ループを構成している各ノード（ノード1と2と3と5）がループを検出した時点で、各ノードがノードごとに設定されたある任意時間のタイマを起動する。最初にノード1がタイマアウトになったため、ノード2と接続されているポートをディセーブルにしてループ経路を断ち切り、バスリセットを発生させる。バスの初期化が完了して、「Branch」ノードから「Leaf」ノードとなったノード1と「Leaf」ノードである4, 6, 7, 8から親（Root）候補であるノードに対してParent notify信号がポートに送信される。「Branch」ノードである2, 3, 5は、Parent notify信号を受信したノードに対して子ノードであることを認めるChild notify信号を送信し、「Leaf」ノードはChild notify信号を受信しそのポートの接続が完了したことを認識する。更に、「Branch」ノード2は、ノード1と接続されていたポートがディセーブルのポートになったため、Parent notify信号を受信していないポートがノード5と接続されているポートだけとなり、そのポートにParent notify信号を送信していることを示している。

【0032】図2は、ノード5が「Root」ノードに確定しツリー識別が完了した状態を示している。この状態に至る際には、まず「Branch」ノード3と5がParent notify信号を受信したポートからChild notify信号を送信してそれぞれが接続を完了する。その後、ノード3からノード5へParent notify信号が送信され、ノード5からノード3へChild notify信号が送信され、ノード5が「Root」ノードに確定して接続が完了する。

【0033】図3は、図1の場合のノード1の代わりにノード5がタイムアウトになってループを検出し、ノード2と接続されているポートをディセーブルにしてバスリセットを発生した場合の図である。バスの初期化が完了した後に、図1と同様にして「Leaf」ノードは親ノードと接続が完了し、「Branch」ノード5は、ノード2と接続されていたポートがディセーブルのポートになったため、Parent notify信号を受信していないポートがノード3と接続されているポートだけとなり、そのポートにParent notify信号を送信して、ノード3がRootになっている。同様に、「Branch」ノード2は、ノード5と接続されていたポートがディセーブルのポートになったため、Parent notify信号を受信していないポートがノ

ード1と接続されているポートだけとなり、そのポートにParent notify信号を送信していることを示している。

【0034】図4は、図3におけるすべてのノードの接続が完了し、ノード3が「Root」ノードに確定し、ツリー識別が完了したことを示している。

【0035】つまり、バス構成はその切断の箇所に応じて変化してしまうが、ループを形成しているノードのうちいずれか一つのノードがループを自動的に切断して再構成することで、ループを回避して正常なバス構成を形成することが可能になる。

【0036】ツリー識別が完了すると、次に各ノードにノード番号を割り振りお互いがデータ通信を可能にするための自己識別に入る。

【0037】＜ツリー識別手順＞次に、本発明の動作フローを図10に示す。

【0038】本動作フローはツリー識別プロセスを実行するフローチャートを示している。この手順はデバイスが接続されているポートからバスリセットを検出した時か、バスリセットを自ら発生した時からスタートし、ツリー識別が完了した時点でエンドとなる。

【0039】まず、バスリセットによってスタートすると、アクティブポートが有るか否かを判別し(S101)、もしアクティブなポートがない場合には、接続されるデバイスがないと認識し、スタンダアロンの動作を開始し(S111)、エンドになる。

【0040】また、アクティブポートが有る場合には、バスリセットを発生するための送信要求か否かを判別し(S102)、もし送信要求の場合にはアクティブなポートすべてにバスリセットを送信し(S103)、バスリセットを送信したアクティブなポートすべてにアイドル信号を送信し(S104)、アクティブポートすべてからアイドル信号受信したか否かを判別し(S105)、アイドル信号を受信したらS201に遷移する。

【0041】ステップS102でバスリセットが送信要求でない場合には、バスリセットを受信し(S106)、アクティブなポートが有るか否かを判別し(S107)、無ければS201へ遷移し、有れば他のアクティブポートすべてにバスリセットを送信し(S108)、バスリセットを送信したアクティブポートすべてにアイドル信号を送信し(S109)、他のアクティブポートすべてからアイドル信号を受信するか否かを判別し(S110)、すべてのアクティブポートからアイドル信号を受信した場合には、S201へ遷移する。

【0042】ここで、上記によって、バスリセットは完了し、ステップS201からは、ツリー識別に入る。

【0043】ステップS201では、タイマ1を起動する。タイマ1は、ツリー識別を完了するには十分な時間であり、タイマ1のタイマアウトはループ接続がありツリー構成が構成できないことを意味している。

【0044】次に、アクティブポートが1ポートか否かを判別し(S203)、1ポートでない場合にはステップS301へ遷移する。アクティブポートが1ポートの場合には、そのポートへParent notify信号を送信し(S203)、タイマ1のタイマアウトか否かを判別し(S204)、タイマアウトの場合には、デバイスが無応答であり通信不能な状態であることを認識し(S216)、そのポートに接続されているデバイスをツリー構成に加えない等のエラー処理を行い(S217)、エンドになる。

【0045】また、ステップS204でタイマ1がタイマアウトしていない場合には、Child notify信号を受信したか否かを判別し(S205)、もし受信している場合には、自デバイスがバス構成中の「Leaf」に確定したことを認識し(S215)、エンドになる。

【0046】もし受信していない場合には、Parent notify信号を受信したか否かを判別し(S206)、もし受信していない場合にはステップS204へ遷移し、もし受信していれば、「Root」を決定するためのRoot競合を開始し、タイマ1を停止した後にタイマ2を起動する(S207)。

【0047】タイマ2は、Root競合のためのタイマで競合した2つのデバイスだけが次にParent notify信号を送信するまでの時間を決めるためのランダムタイマである。

【0048】ここで次にタイマ2がタイマアウトしたか否かを判別し(S208)、タイマアウトになった場合には、Parent notify信号を受信したか否かを判別し(S209)、もし受信した場合には、Child notify信号を送信し(S213)、自デバイスがバス構成中の「Root」に確定したことを認識しエンドになる。

【0049】ステップS209でParent notify信号を受信しなかった場合には、Parent notify信号を送信し、次にそのポートからChild notify信号を受信したか否かを判別し(S211)、受信した場合には、自デバイスがバス構成中の「Leaf」に確定したことを認識しエンドになる。次に、ステップS202でアクティブなポートが複数ポート有る場合には、そのアクティブなポート数をNに設定し(S301)、タイマ1のタイムアウトが発生したか否かを判別し(S302)、もしタイマアウトが発生していない場合には、NのいずれかのポートからParent notify信号を受信したか否かを判別し(S303)、もし受信していない場合にはS302へ遷移し、もし受信していれば、Parent notify受信したポートへchild notifyを送信し(S304)、Nを1ポート分減少させ(S305)、Nが1ポートになったか否かを判別し(S306)、もし1ポートでない場合(つまり2ポート以上)には、S302へ遷移し、もし1ポートになった場合には、S501へ遷移する。

【0050】次に、ステップS302でタイマ1のタイ

ムアウトが発生した場合には、無応答の状態が続いているポートが存在するためループが発生したと認識して、通常であればここでループ検出で終了してしまうが、本発明にかかるノードでは、次に、ループの回避のプロセスに入る。まず、タイマ3を起動する(S401)。

【0051】タイマ3は、ループを形成しているデバイスすべてが対象となるが、各デバイスで固有のタイマ値をもつことが必要であるが、そのタイマ値についてはループを形成しているポート数の多いデバイスはより長いタイマ値をもつようにすること、ループを形成しているポート数が最も少ない複数のデバイスの中から唯一のデバイスがループ解除を実行してバスリセットを発生させるように各デバイスのタイマ値を設定するようにする。

【0052】例えば、下記のようなタイマ値にすることで実現が可能になる。

【0053】 $T3 = T\alpha + T\beta \times N + T\gamma$

$T\alpha$: 任意の固定タイマ値

$T\beta$: 無応答のポートが存在する場合に適応される固定タイマ値(このタイマ値に無応答のポート数Nが乗じられる)

$T\gamma$: デバイスによって異なるタイマ値(例えば、カンパニIDやユニークID等デバイス固有の値を用いてデバイス固有のタイマ値を算出する)

ここで、 $T\beta > T\gamma$ とする。

【0054】次に、タイマ3がタイムアウトするか否かを判別し(S402)、もし発生していない場合には、NのいずれかのポートからParent notify信号を受信しているか否かを判別し(S403)、もし受信していない場合には、S406へ遷移し、もし受信している場合には、Parent notify信号を受信してポートへChild notify信号を送信し(S404)、Nを1ポート分減少し(S405)、次に、Nのいずれかのポートからアクティブでないポートを検出したか否かを判別し(S406)、もし検出していない場合にはS408へ遷移し、検出した場合には、Nを1ポート分減少し(S407)、次にNが1ポートになったか否かを判別し(S408)、もし1ポートでない場合(つまり2ポート以上)には、S402へ遷移し、もし1ポートになった場合には、S501へ遷移する。

【0055】次に、ステップS402でタイマ3のタイムアウトが発生した場合には、自デバイスがループ解除処理を実行するデバイスであることを認識し、N個のポートのうち、いずれかひとつのポートをディセーブルにし(S409)、バスリセットを発生を要求し(S410)、ステップS101へ遷移する。

【0056】ステップS501では、残りのポートへParent notify信号を送信し、タイマ1のタイムアウトか否かを判別し(S504)、タイムアウトの場合には、デバイスが無応答であり通信不能な状態であることを認識し(S514)、そのポートに接続されているデバイ

スをつリー構成に加えない等のエラー処理を行い(S515)、エンドになる。

【0057】また、ステップS502でタイマ1がタイムアウトしていない場合には、Child notify信号を受信したか否かを判別し(S503)、もし受信している場合には、自デバイスがバス構成中の「Branch」に確定したことを認識し(S513)、エンドになる。

【0058】もし受信していない場合には、Parent notify信号を受信したか否かを判別し(S504)、もし受信していない場合にはS502へ遷移し、もし受信していれば、「Root」を決定するためのRoot競合を開始し、タイマ1を停止した後にタイマ2を起動する(S505)。

【0059】次にタイマ2がタイムアウトしたか否かを判別し(S506)、タイムアウトになった場合には、Parent notify信号を受信したか否かを判別し(S507)、もし受信した場合には、Child notify信号を送信し(S511)、自デバイスがバス構成中の「Root」に確定したことを認識しエンドになる。

【0060】ステップS507で、Parent notify信号を受信しなかった場合には、Parent notify信号を送信し、次にそのポートからChild notify信号を受信したか否かを判別し(S211)、受信した場合には、自デバイスがバス構成中の「Branch」に確定したことを認識しエンドになる。

【0061】図15~17は、上記動作フローに基づいて、ループ接続が複数存在するバス構成におけるループ回避手順の一例を示している。

【0062】図15(a)は、ノード10、11、12、13のすべてのポートをループ接続した状態を示している。この場合には、各ノードともすべてのポートがループ接続になっているため、すべてのポートからParent notify信号が送信されず、また、すべてのポートにParent notify信号が受信されず接続は完了しないためループが検出される。そこで、各ノードがループを検出すると、各ノードが任意のタイマを起動する。ここで、すべてのノードが3ポート共ループ検出しているので、すべてのノードがループを検出したポート数が3となり、ポート数によるタイマ値 $T\beta$ は同じタイマ値になるの

で、各ノード固有のタイマ値 $T\gamma$ により、いずれか一つのノードが最初にタイムアウトになる。

【0063】図15(b)は、ノード10かノード13のどちらかが先にタイムアウトになり、ノード10がタイムアウトになった場合には、ノード13に接続されているポートをディセーブルにしてバスリセットを発生させ、ノード13がタイムアウトになった場合には、ノード10に接続されているポートをディセーブルにしてバスリセットを発生させる。そこで、バスの初期化が完了した後、ノード10とノード13の残り2ポートとノード11とノード12の3ポートで再びループが検出さ

れ、再び、各ノードとも任意のタイマ値を起動する。

【0064】ここで、ループを検出したポート数は、ノード10とノード13では2ポート、ノード11とノード12では3ポートであるので、ポート数によるタイマ値 T_B はノード11とノード12よりノード10とノード13の方がポート数が少ない分短いタイマ値になり、より早くタイマアウトになるが、ノード固有のタイマ値 T_A により、ノード10とノード13のうちのどちらが先にタイマアウトになる。

【0065】図16(a)は、ノード13が先にタイマアウトになり、ノード11に接続されているポートをディセーブルにしてバスリセットを発生させる。そこで、バスの初期化が完了した後、ノード13はポートが1ポートとなり「Leaf」ノードとなりノード12と接続されているポートは接続が完了するが、ノード10とノード11とノード12の残り2ポートでは再びループが検出され、再び、このループを検出したノードは任意のタイマ値を起動する。

【0066】ここで、ループを検出したノードはすべて2ポートで検出しているので、ループを検出したポート数が2となり、ポート数によるタイマ値 T_B は同じタイマ値になるので、各ノード固有のタイマ値 T_A により、いずれか一つのノードが最初にタイムアウトになる。

【0067】図16(b)は、ノード12が先にタイマアウトになった場合の例を示しており、ノード10に接続されているポートをディセーブルにしてバスリセットを発生させる。そこで、バスの初期化が完了した後、ノード10とノード11とノード12とノード13がデジチェーン接続された正常なバス構成が形成されることでツリー識別が完了し、その後自己識別を行い、データ通信が可能になる。

【0068】図17(a)は、ノード12が先にタイマアウトになった場合の例を示しており、図16(b)とは違うノード11に接続されているポートをディセーブルにしてバスリセットを発生させる。そこで、バスの初期化が完了した後、ノード11とノード10とノード12とノード13がデジチェーン接続された正常なバス構成が形成されることでツリー識別が完了し、その後自己識別を行い、データ通信が可能になる。

【0069】図17(b)は、ノード10が先にタイマアウトになった場合の例を示しており、ノード11に接続されているポートをディセーブルにしてバスリセットを発生させる。そこで、バスの初期化が完了した後、ノード12がノード11とノード10とノード13とブランチ接続された正常なバス構成が形成されるとツリー識別が完了し、その後自己識別を行い、データ通信が可能になる。

【0070】このように、複数のループが存在する場合にもループを回避して正常なバス構成を形成することが可能になる。

【0071】以上説明したように、本発明にかかるIEEE 1394インターフェースを備えたノードは、ループを検出してそれを解除することができる。ループを解除する際には、各ノードの有するポートの数及び各ノードに固有の時間により、ループを検出してからポートをディセーブルにするまでの時間をノードごとに変えている。こうすることで、1回のループ解除処理でディセーブルになるリンクは1カ所に限られ、ネットワークを分断するという事態の発生を防止できる。

【0072】

【他の実施形態】なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0073】また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを

読み出し実行することによっても達成される。

【0074】この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0075】プログラムコードを供給するための記憶媒体としては、例えば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモ리카ード、ROMなどを用いることができる。

【0076】また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0077】さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0078】

【発明の効果】以上説明したように、バス構成を形成できないループが検出された際に、自動的にそのループを回避して正常なバス構成を形成でき正常なデータ通信が可能になるため、バス構成の形成時の信頼性が飛躍的に

向上するという効果がある。

【0079】また、操作者がループを形成しているケーブルを外すという操作が不要になるため、操作者の利便性も飛躍的に向上するという効果がある。

【図面の簡単な説明】

【図1】ノードAによるループ切断を示す図である。

【図2】ノードAによるループ切断後のツリー完了を示す図である。

【図3】ノードEによるループ切断を示す図である。

【図4】ノードEによるループ切断後のツリー完了を示す図である。

【図5】各ノードの構成例を示す図である。

【図6】従来例(1)を示す図である。

【図7】従来例(2)を示す図である。

【図8】従来例(3)を示す図である。

【図9】従来例(4)を示す図である。

【図10】本発明にかかるノードの動作フロー(1)を示す図である。

【図11】本発明にかかるノードの動作フロー(2)を示す図である。

【図12】本発明にかかるノードの動作フロー(3)を示す図である。

【図13】本発明にかかるノードの動作フロー(4)を示す図である。

【図14】本発明にかかるノードの動作フロー(5)を示す図である。

【図15】本発明にかかるループ切断の様子を示す他の図である。

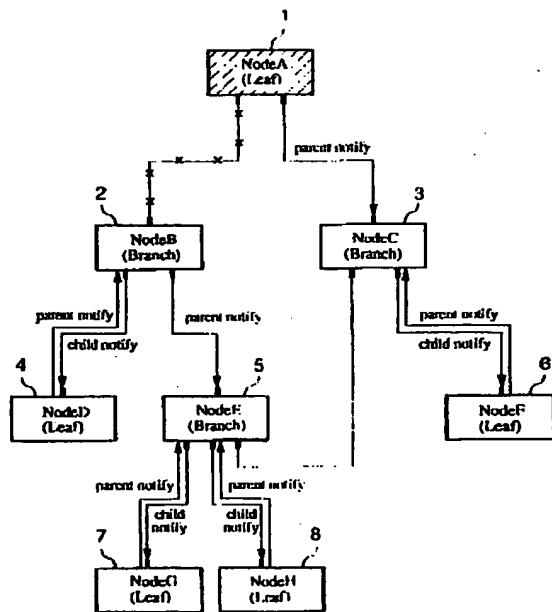
【図16】本発明にかかるループ切断の様子を示す他の図である。

【図17】本発明にかかるループ切断の様子を示す他の図である。

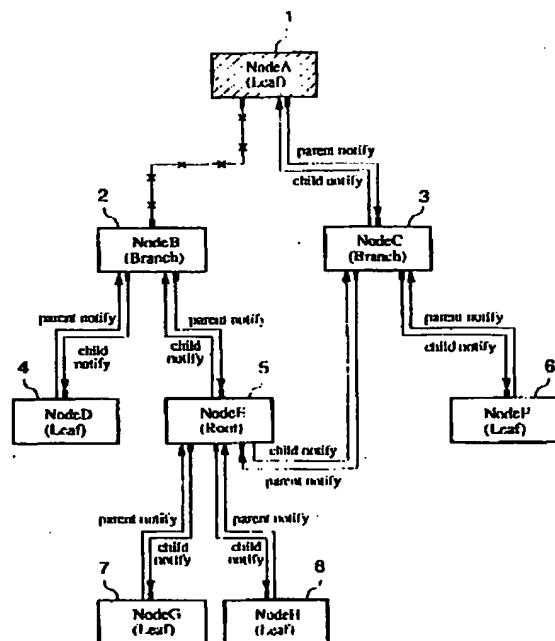
【符号の説明】

1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13 ノード

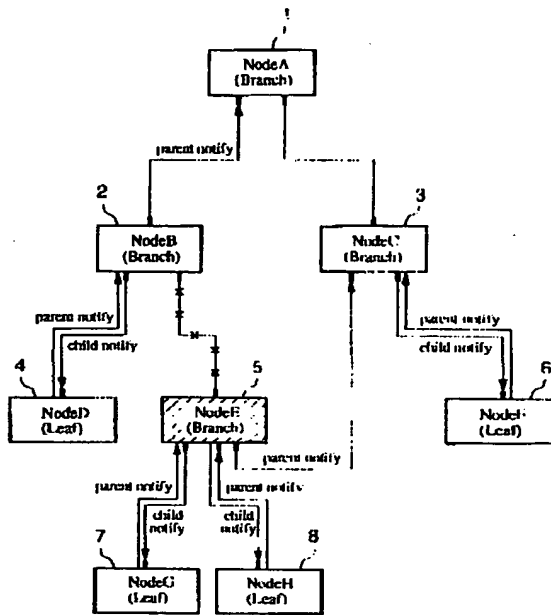
【図1】



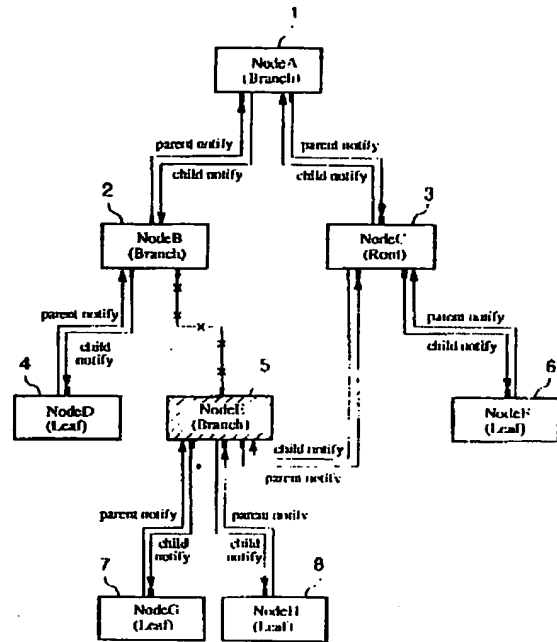
【図2】



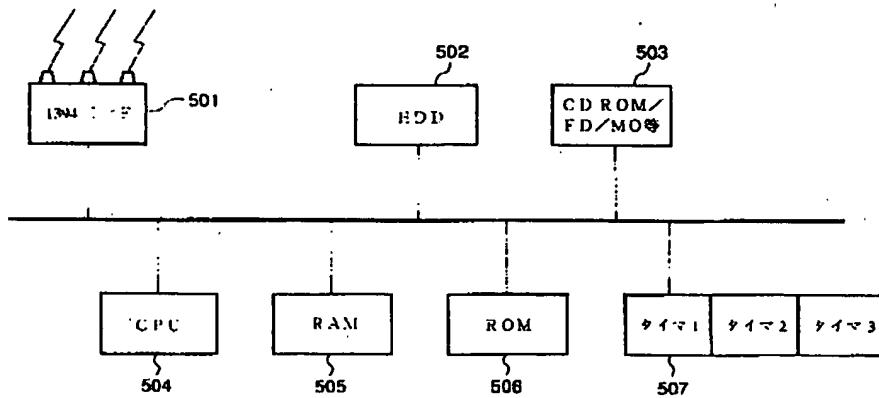
【図3】



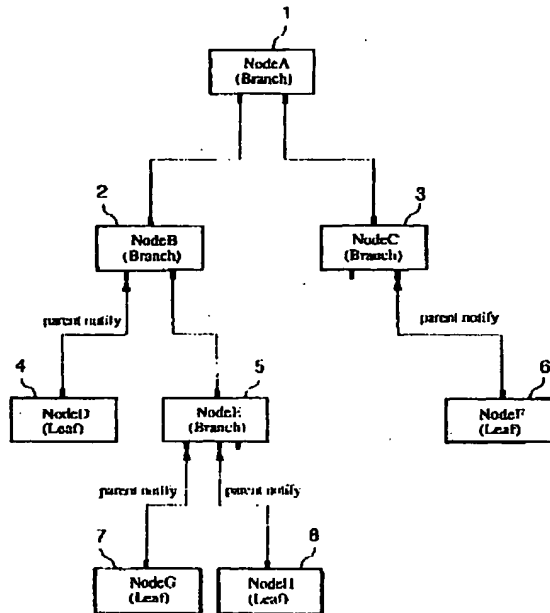
【図4】



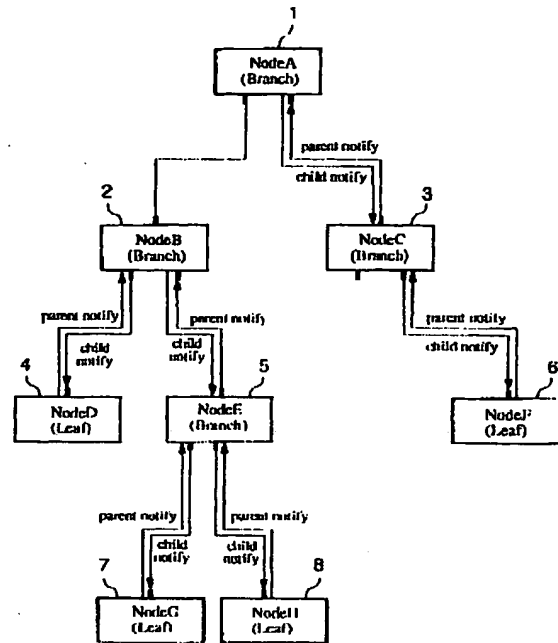
【図5】



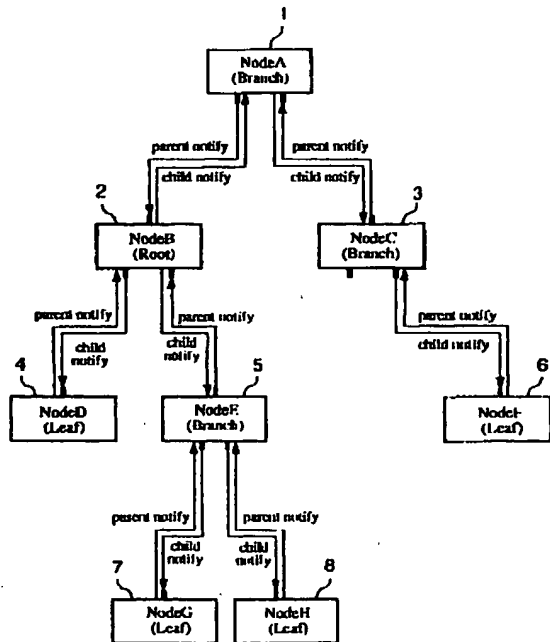
【図6】



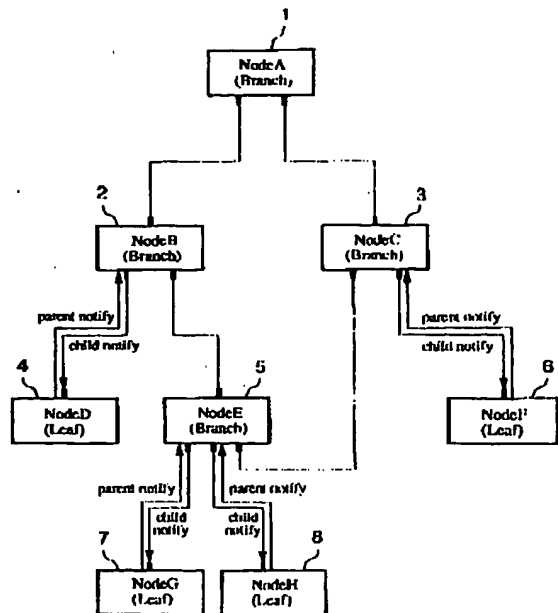
【図7】



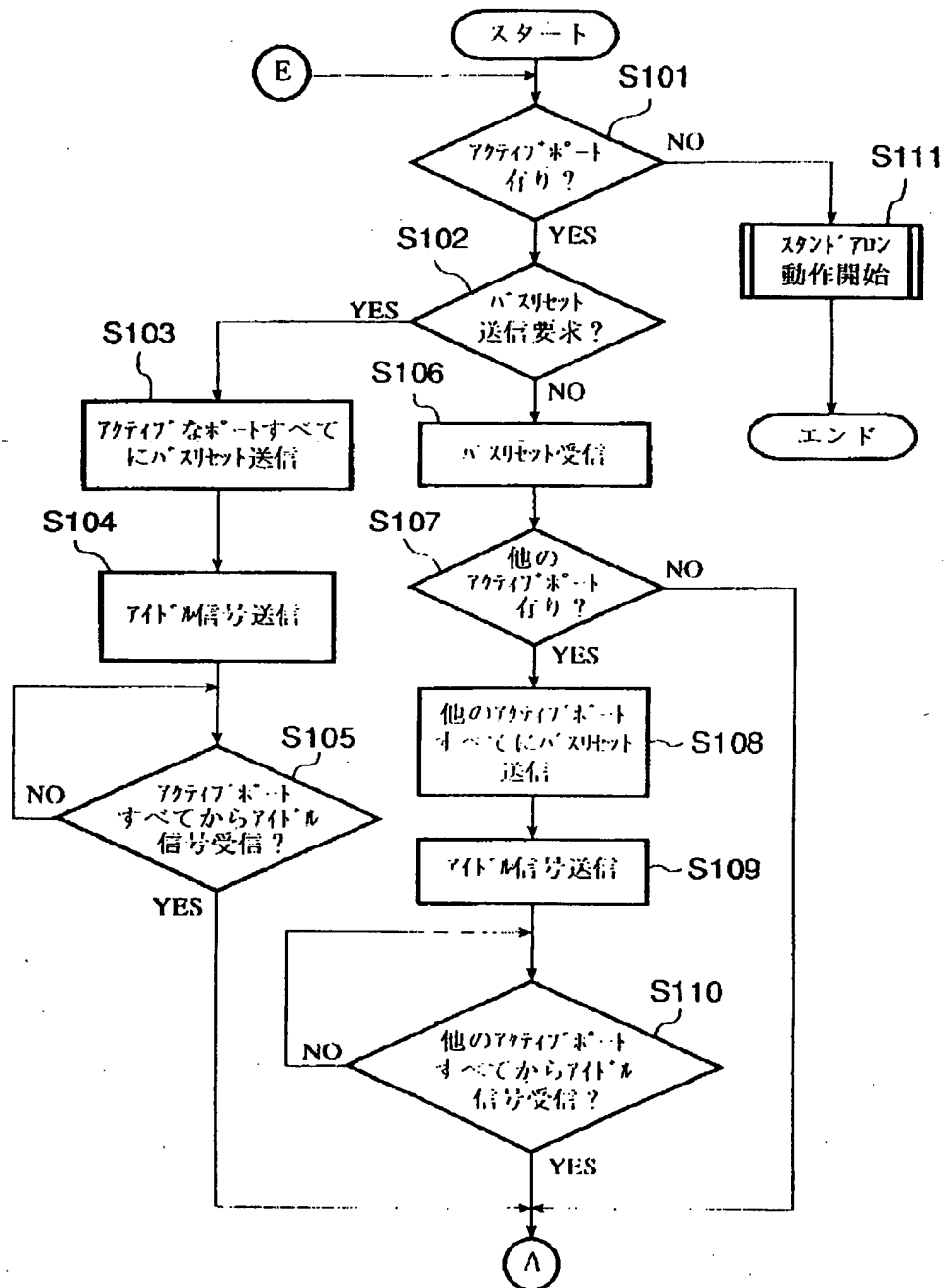
【図8】



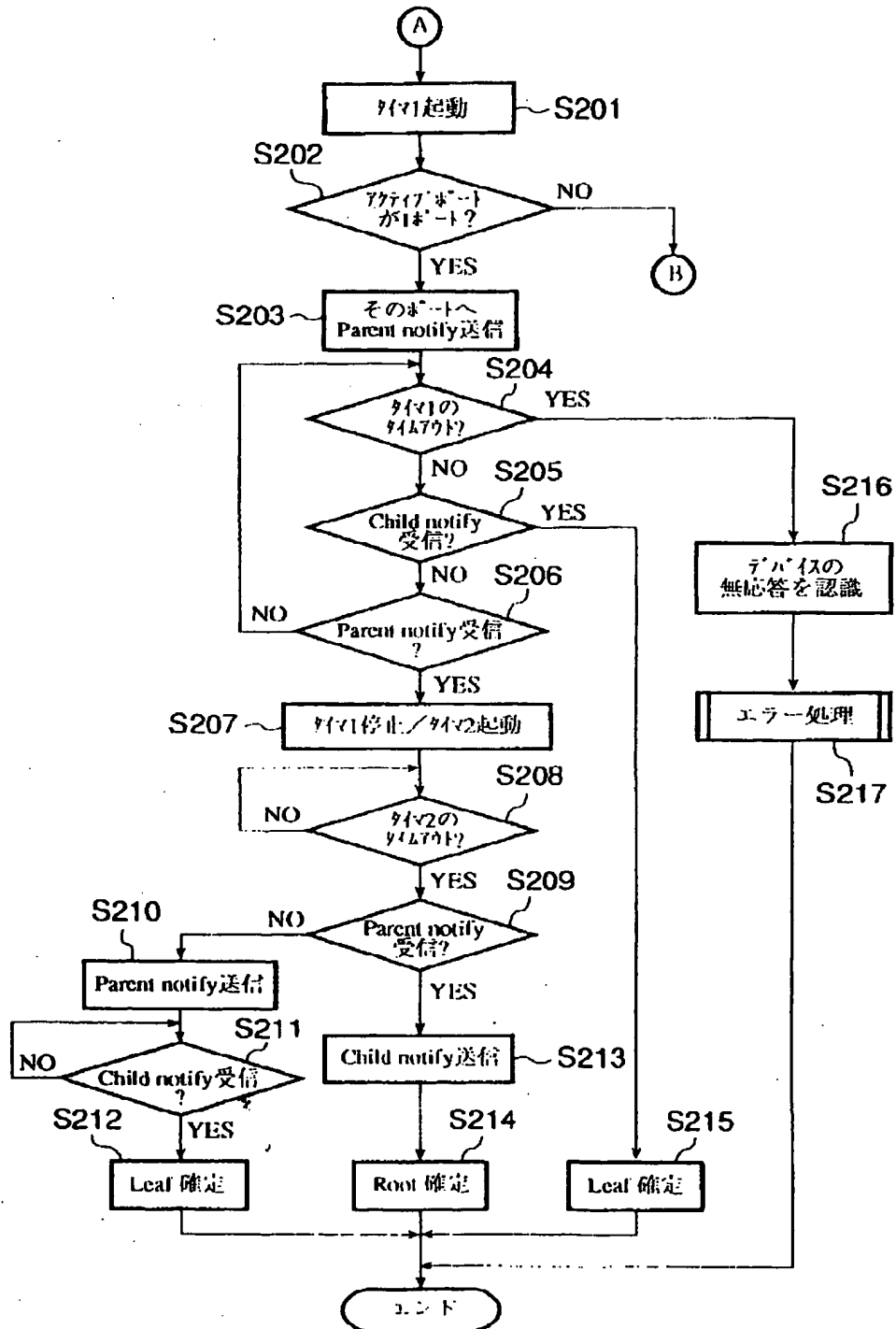
【図9】



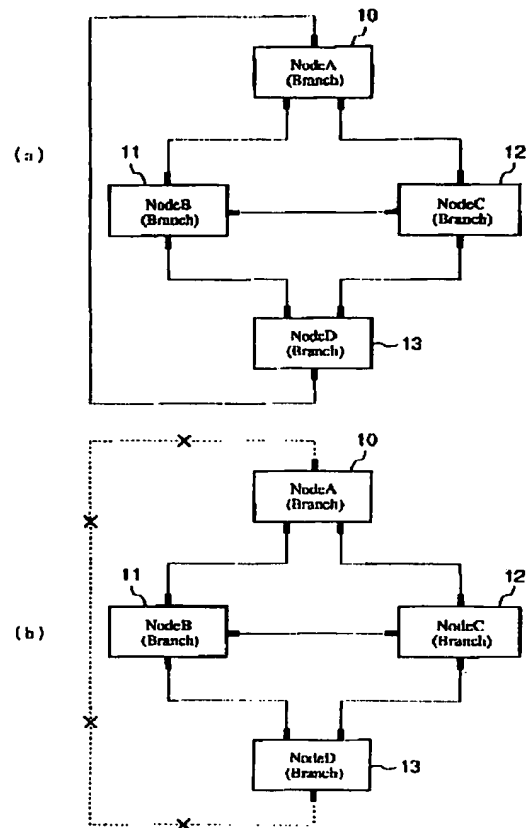
【図10】



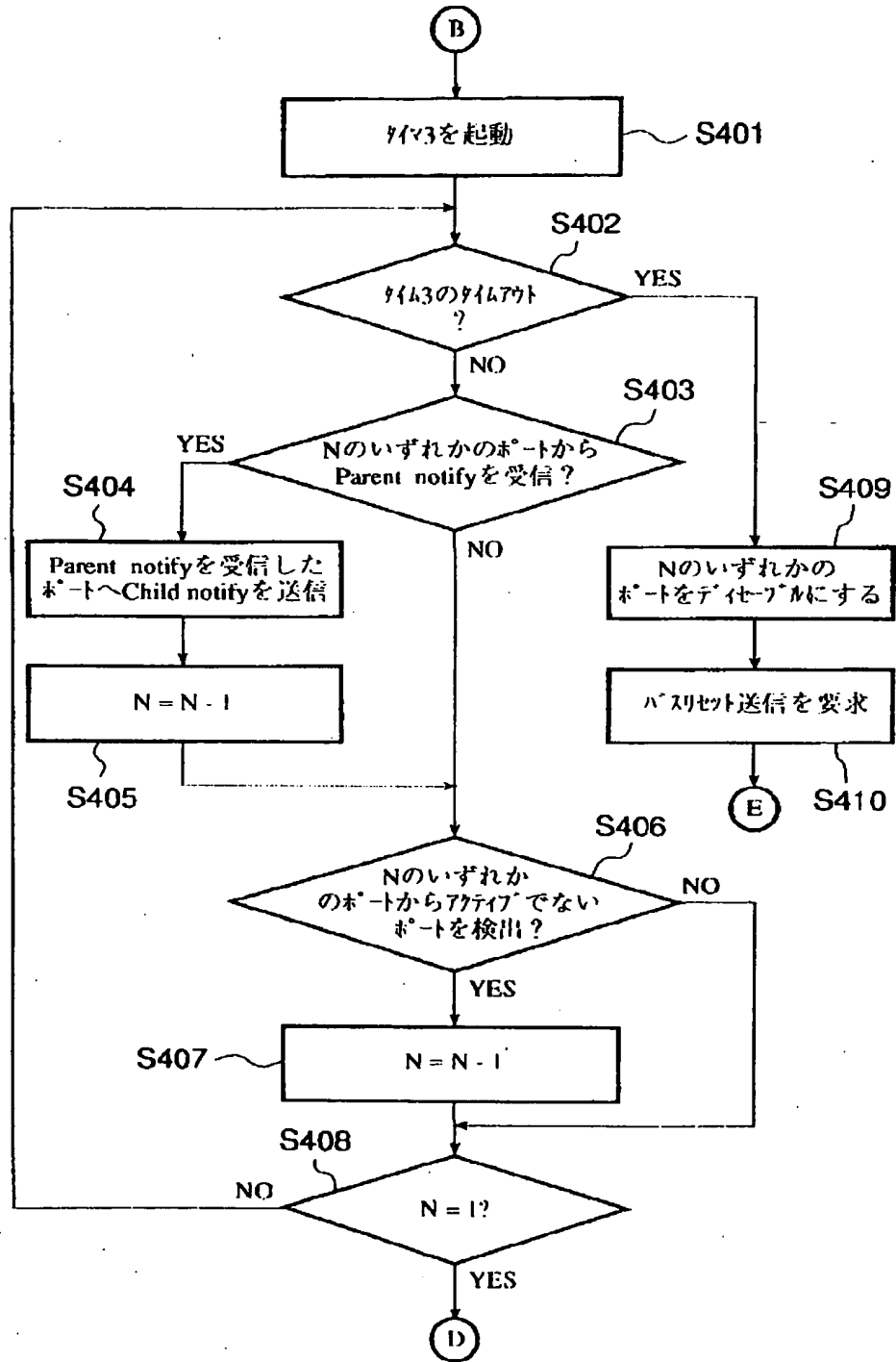
【図11】



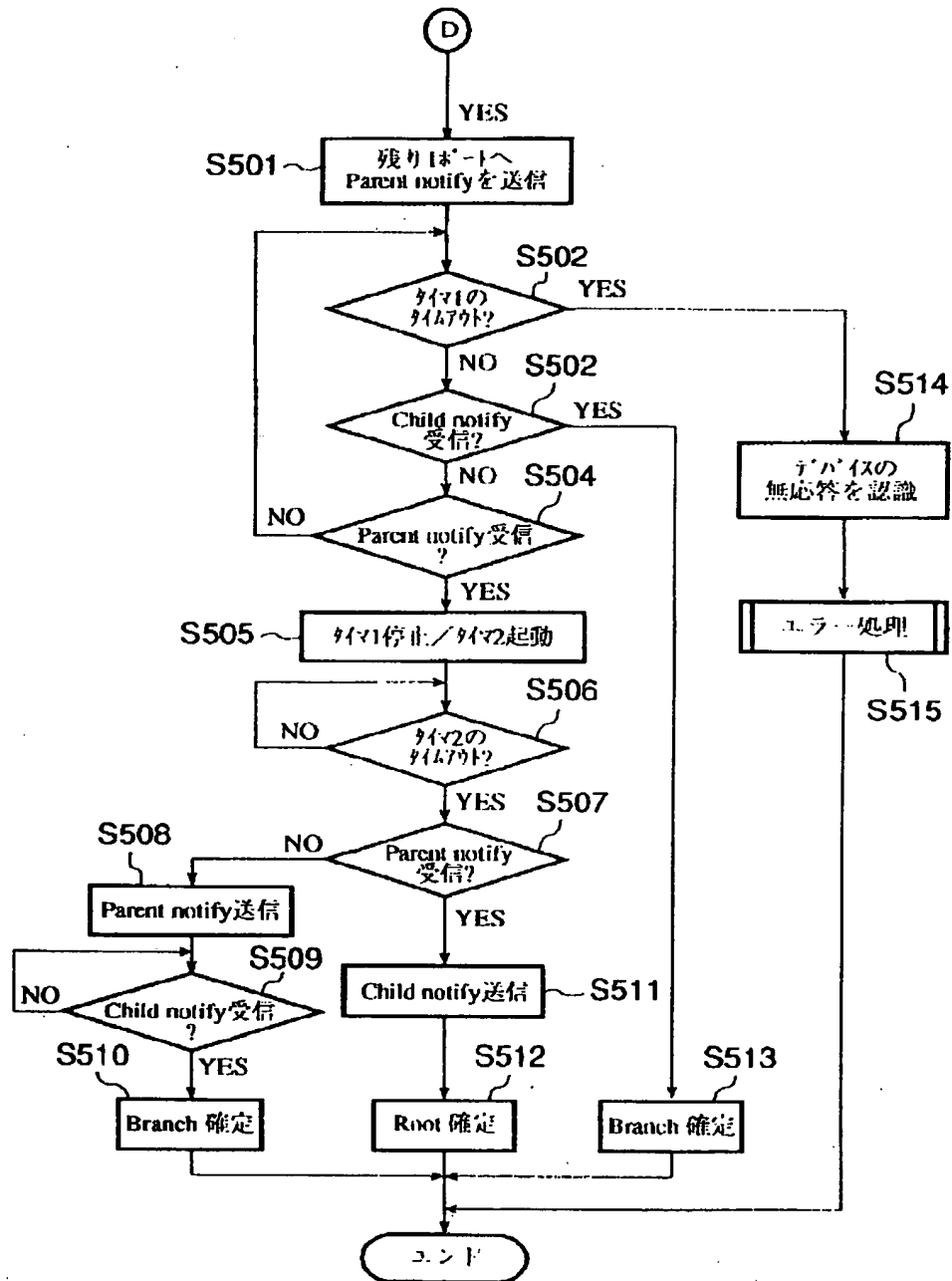
【图 1.5】



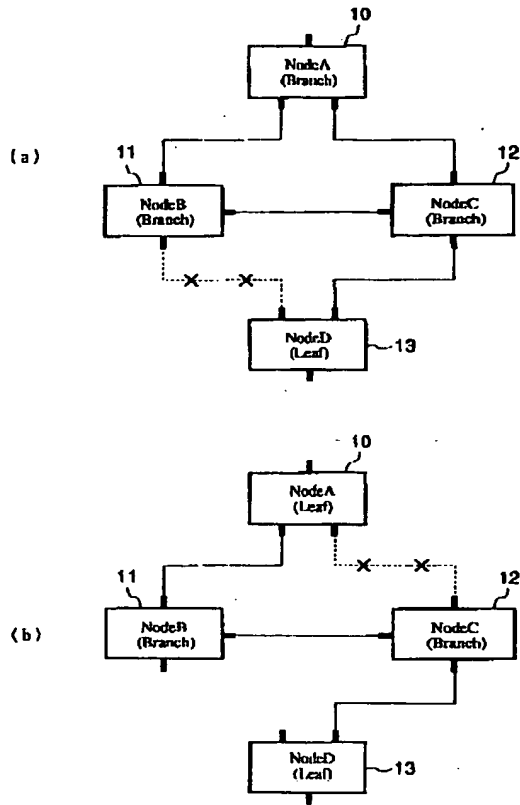
【図13】



【図14】



【図16】



【図17】

